

Mesaninen – Spring

Kompendium

Contents

Mesaninen – Spring	1
Kompendium	1
1. Dependency Injection.....	2
1.1 XML.....	2
1.1.1 Instansiering av bønner	2
1.1.2 Injection av bønner	2
1.1.3 Injection av verdier	3
1.2. Annotasjon	4
1.2.1 Instansiering av bønner	4
1.2.2 Injection av bønner	4
1.2.3 Injection av bønner (navngitt)	4
1.2.4 Injection av verdier	4
1.2.5 Injection av verdier (fra properties)	4
2. Aspektorientering.....	5
3. Transaksjoner	5
4. Lifecycle-annotasjoner	5

1. Dependency Injection

1.1 XML

1.1.1 Instansiering av bønner

```
public class Kniv {  
  
}
```

```
<bean id="kniv" name="kniv" class="no.mesan.spring.Kniv" />
```

Det er kun class som er påkrevd, har man kun en bønne per klasse (singletons) så holder det å angi klasse.

1.1.2 Injection av bønner

1.1.2.1 Constructor-arg

```
public class Kokk {  
  
    private final Kniv kniv;  
  
    public Kokk(final Kniv kniv) {  
        this.kniv = kniv;  
    }  
}
```

```
<bean id="kokk" name="kokk" class="no.mesan.spring.Kokk">  
    <constructor-arg name="kniv" ref="kniv" />  
</bean>
```

1.1.2.2 Property

```
public class Kokk {  
  
    private Kniv kniv;  
  
    public void setKniv(final Kniv kniv) {  
        this.kniv = kniv;  
    }  
}
```

```
<bean id="kokk" name="kokk" class="no.mesan.spring.Kokk">  
    <property name="kniv" ref="kniv" />  
</bean>
```

1.1.3 Injection av verdier

1.1.3.1 Constructor-arg

```
public class Kokk {  
    private final String navn;  
  
    public Kokk(final String navn) {  
        this.navn = navn;  
    }  
}
```

```
<bean id="kokk" name="kokk" class="no.mesan.spring.Kokk">  
    <constructor-arg name="navn" value="Eyvind Hellstrøm" />  
</bean>
```

1.1.3.2 Property

```
public class Kokk {  
    private String navn;  
  
    public void setNavn(final String navn) {  
        this.navn = navn;  
    }  
}
```

1.2. Annotasjon

1.2.1 Instansiering av bønner

```
<context:component-scan base-package="no.mesan.spring" />
```

```
@Component
public class Kniv {

}
```

1.2.2 Injection av bønner

```
@Component
public class Kokk {
    @Inject
    private Kniv kniv;
}
```

1.2.3 Injection av bønner (navngitt)

```
@Component
public class Kokk {
    @Inject
    @Named("kniv")
    private Kniv kniv;
}
```

1.2.4 Injection av verdier

```
@Component
public class Kokk {
    @Value("Gordon Ramsey")
    private String navn;
}
```

1.2.5 Injection av verdier (fra properties)

```
<context:property-placeholder location="classpath:spring.properties" />
```

```
kokk.navn=Gordon Ramsey
```

```
@Component
public class Kokk {
    @Value("${kokk.navn}")
    private String navn;
}
```

Man benytter samme notasjon med \${} og property-placeholder ved xml-konfigurasjon.

2. Aspektorientering

```
<bean class="no.mesan.spring.Aspekt" />

<bean class="no.mesan.spring.KlasseSomBrukerAspekt" />

<aop:aspectj-autoproxy/>
```

1. Bønne for selve aspektet
2. Bønne for klasse som skal bruke aspektet
3. «Skrå på» aspektorientering

3. Transaksjoner

```
<tx:annotation-driven />
```

```
@Transactional(readOnly = true)
public String lesNoe() {
```

```
@Transactional
public void skrivNoe(final String noe) {
```

4. Lifecycle-annotasjoner

```
@PostConstruct
public void postConstruct() {
```

```
@PreDestroy
public void preDestroy() {
```